Practical 8: Using JFLAP, create SLR(1) parse table for a given grammar. Simulate parsing and output the parse tree in proper format.

Accept the input string with Regular expression of Finite Automaton: 101+.

Source code:

```
def FA(s):
#if the length is less than 3 then it can't be accepted, Therefore end the
process.
   if len(s):
        return "Rejected"
  #first three characters are fixed. Therefore, checking them using index
    if s[0]=='1':
       if s[1]=='0':
          if s[2]=='1':
             # After index 2 only "1" can appear. Therefore break the process if any other
  character is detected
             for i in range(3,len(s)):
                if s[i]!='1':
                   return "Rejected"
             return "Accepted" # if all 4 nested if true
          return "Rejected" # else of 3rd if
       return "Rejected" # else of 2nd if
    return "Rejected" # else of 1st if
  inputs=['1','10101','101','10111','01010','100','','10111101','1011111']
  for i in inputs:
    print(FA(i))
```

Output:

```
Rejected
Rejected
Accepted
Accepted
Rejected
Rejected
Rejected
Rejected
Accepted
```